Intellectual Property Administration
P. O. Box 272400
Fort Collins, Colorado 80527-2400

PATENT APPLICATION

ATTORNEY DOCKET NO. 200301967-2

IN THE
UNITED STATES PATENT AND TRADEMARK OFFICE

Inventor(s): Andrej Kocev et al. **Confirmation No.:** 1813

Application No.: 09/944,776 **Examiner:** Pham, Thomas K.

Filing Date: 08/31/2001 **Group Art Unit:** 2121

Title: PROGRAMMABLE TUNING FOR FLOW CONTROL AND SUPPORT FOR CPU HOT PLUG

Mail Stop Appeal Brief-Patents
Commissioner For Patents
PO Box 1450
Alexandria, VA 22313-1450

TRANSMITTAL OF APPEAL BRIEF

Sir:

Transmitted herewith is the Appeal Brief in this application with respect to the Notice of Appeal filed on 12/7/2005.

The fee for filing this Appeal Brief is (37 CFR 1.17(c)) $500.00.

(complete (a) or (b) as applicable)

The proceedings herein are for a patent application and the provisions of 37 CFR 1.136(a) apply.

( ) (a) Applicant petitions for an extension of time under 37 CFR 1.136 (fees: 37 CFR 1.17(a)-(d) for the total number of months checked below:

( )	one month	$120.00
( )	two months	$450.00
( )	three months	$1020.00
( )	four months	$1590.00

( ) The extension fee has already been filled in this application.

(X) (b) Applicant believes that no extension of time is required. However, this conditional petition is being made to provide for the possibility that applicant has inadvertently overlooked the need for a petition and fee for extension of time.

Please charge to Deposit Account **08-2025** the sum of **$0.00**. At any time during the pendency of this application, please charge any fees required or credit any over payment to Deposit Account 08-2025 pursuant to 37 CFR 1.25. Additionally please charge any fees to Deposit Account 08-2025 under 37 CFR 1.16 through 1.21 inclusive, and any other sections in Title 37 of the Code of Federal Regulations that may regulate fees. A duplicate copy of this sheet is enclosed.

(X) I hereby certify that this correspondence is being deposited with the United States Postal Service as first class mail in an envelope addressed to: Commissioner for Patents, Alexandria, VA 22313-1450. Date of Deposit: 02/06/2006

OR

( ) I hereby certify that this paper is being transmitted to the Patent and Trademark Office facsimile number _____ on _____

Number of pages: 27

Typed Name: Melissa Altman

Signature: Melissa Altman

Respectfully submitted,

Andrej Kocev et al.

By _____

Michael R. Reinemann

Attorney/Agent for Applicant(s)
Reg. No. 38,280

Date: 02/06/2006

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In Re The Application of:
Andrej Kocev et al.

Serial No.: 09/944,776

Filed: August 31, 2001

For: Programmable Tuning for Flow Control and Support for CPU Hot Plug

Examiner: Pham, Thomas K.

Art Unit: 2121

Cesari and McKenna, LLP
88 Black Falcon Avenue
Boston, MA 02210
February 6, 2006

CERTIFICATE OF MAILING

I hereby certify that the following paper is being deposited with the United States Postal Service as first-class mail in an envelope addressed to the Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450, on February 6, 2006.

Melissa Altman

Mail Stop Appeal Brief-Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

APPEAL BRIEF

In response to the Notice of Reinstatement of Appeal mailed December 7, 2005,

Applicants hereby submit this new Appeal Brief.

REAL PARTY IN INTEREST

The real party in interest is Hewlett-Packard Development Company, L.P. of Houston, Texas.

RELATED APPEALS AND INTERFERENCES

Applicants and their legal representatives know of no related appeals or interferences that will directly affect or be directly affected by or have a bearing on the Board's decision in the present appeal.

STATUS OF CLAIMS

Claims 1-12 have been canceled. Claims 13-41 are pending in the case. Claims 13-36 and 40-41 stand rejected under 35 U.S.C. §103. Dependent claims 37-39 are objected to, but would be allowable if rewritten in independent form including all of the limitations of the base claim and any intervening claims. As indicated below, claim 37 has been rewritten in independent form.

A copy of claims 13-41, in their current form, is attached hereto as an Appendix.

STATUS OF AMENDMENTS

Applicants filed an Amendment After Appeal on the same date as the filing of this new Appeal Brief. The Amendment After Appeal rewrites dependent claim 37 in independent form including all of the limitations of the base claim and intervening claims, pursuant to 37 C.F.R. §41.33(b)(2). Dependent claims 38-39 depend from claim 37 and therefore have not been amended.

SUMMARY OF CLAIMED SUBJECT MATTER

The summary is set forth in four exemplary embodiments that correspond to independent claims 13, 15, 21 and 32. Discussions about elements and recitations of these claims can be found at least at the cited locations in the specification and drawings.

Independent Claim 13 is directed to a method for programmatically allocating system resources to accommodate input/output (I/O) transactions at I/O ports of a multiprocessor computer. The claimed method includes determining the number of devices being serviced by the I/O ports (see pp. 3 and 9-10), identifying an assembly for hot swapping (see p. 11), copying the contents of cache memories associated with the assembly to be hot swapped (see pp. 3 and 12-13), setting criteria for transactions at the I/O ports with regard to the number of devices determined to be serviced thereby (see p. 10), and assigning resources to the I/O ports with respect to the number of devices determined to be serviced thereby (see p. 10).

Independent claim 15 is directed to a system for allocating resources to accommodate I/O transactions at the I/O ports of a multiprocessor computer. The system includes means for determining the number of devices being serviced by the I/O ports (see pp. 3 and 9-10, and Figs. 7-8), an assembly for hot swapping (see pp. 5-6 and 11, and Fig. 3), means for copying the contents of cache memories associated with the assembly to be hot swapped (see pp. 3 and 13-13, and Figs. 7-8), means for setting criteria for transactions at the I/O ports with regard to the number of devices determined to be serviced thereby (see p. 10 and Figs. 7, 8 and 10B), and means for assigning resources to the I/O

ports, whereby the assigning means are responsive to the criteria (see p. 10 and Figs. 7, 8 and 10B).

Independent claim 21 is directed to a method for programmatically allocating resources for processing I/O transactions at the I/O ports of an I/O bridge. The claimed method includes identifying the number of I/O devices being serviced by an I/O port (see pp. 3 and 9-10), setting criteria for the transactions at the I/O port based upon the number of devices that it services (see p. 10), and assigning resources to the I/O port in response to the criteria that have been set (see p. 10).

Independent claim 32 is directed to an I/O bridge for use in a multiprocessor computer. The I/O bridge includes a plurality of I/O ports for communicating with I/O devices that generate or receive transactions (see p. 6, and Figs. 4 and 7), resources for use in servicing those transactions (see pp. 9-10, and Fig. 8), and programmable logic that assigns the resources among the I/O ports in response to the number of I/O devices to which the I/O ports are communicating (see pp. 10-12, and Figs. 10-12).

GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

Whether claims 13, 15 and 21, which otherwise meet all conditions of patentability under Title 35 of the United States Code, are unpatentable under 35 U.S.C. §103 over U.S. Patent No. 6,119,185 to Westerinen et al. ("Westerinen") in view of U.S. Pat. No. 6,718,413 to Wilson ("Wilson") and U.S. Patent No. 6,219,734 to Wallach et al. ("Wallach"), where the art of record either alone or in combination fails to teach or suggest, among other things, setting criteria for transactions at a port with regard to the number of

devices serviced by the port, assigning resources to the port based on the number of devices being serviced, or determining the number of devices being serviced by the ports.

Whether claim 21, which otherwise meet all conditions of patentability under Title 35 of the United States Code, is unpatentable under 35 U.S.C. §103 over Westerinen in view of Wilson, where the art of record either alone or in combination fails to teach or suggest, among other things, setting criteria for transactions at a port with regard to the number of devices serviced by the port, or assigning resources to the port based on the number of devices being serviced.

Whether claims 14, 16 and 31, which otherwise meet all conditions of patentability under Title 35 of the United States Code, are unpatentable under 35 U.S.C. §103 over Wallach, where Wallach fails to teach or suggest, among other things, assigning one of control registers, direct memory access engines or cache memories to ports.

Whether claims 17 and 19, which otherwise meet all conditions of patentability under Title 35 of the United States Code, are unpatentable under 35 U.S.C. §103 over Westerinen, where Westerinen fails to teach or suggest, among other things, determining the number and types of transactions anticipated at the ports or the assignment of resources with respect to the number and types of transactions.

Whether claim 32, which otherwise meet all conditions of patentability under Title 35 of the United States Code, is unpatentable under 35 U.S.C. §103 over Westerinen, where Westerinen fails to teach or suggest, among other things, programmable logic configured and arranged to assign resources among I/O ports in response to the number of I/O devices with which the I/O ports are communicating.

ARGUMENT

Legal Standard

In rejecting claims under 35 U.S.C. §103, the examiner bears the initial burden of presenting a prima facie case of obviousness. See, e.g., In re Rijckaert, 9 F.3d 1531, 1532 (Fed. Cir. 1993). To establish a prima facie case of obviousness, the references, when considered in their entirety, must teach or suggest all of the claimed limitations. If the references fail to teach or suggest any one of the claimed limitations, then the rejection should be reversed. An examiner may not, moreover, resort to speculation, unfounded assumption or hindsight reconstruction to supply deficiencies in the references. In re Warner, 379 F.2d 1011, 1017 (CCPA 1967). In addition, the Examiner must set forth specific reasons why one skilled in the art would be motivated to select the cited features for combination in the manner claimed. In re Rouffet, 149 F.3d 1350, 1357 (Fed. Cir. 1998). Such a requirement is intended to avoid impermissible hindsight, in which the inventor's own disclosure is used as a blueprint for piecing together the prior art to defeat patentability. In re Dembiczak, 175 F.3d 994, 999 (Fed. Cir. 1999).

The claims do not stand or fall together. Instead Applicants present separate arguments for various independent and dependent claims. Each of these arguments is separately argued below and presented with separate headings and sub-headings as required by 37 C.F.F. §41.37(c)(1)(vii).

Claims 13, 15 and 21

Independent claim 13 recites, in relevant part, as follows:

> "A method for programmably allocating resources to accommodate I/O transactions at I/O ports of a multiprocessor computer system comprising:"

"determining the number of devices being serviced via the ports",

"setting criteria **for transactions** at the port **with respect to the number of devices**", and

"with respect to the numbers of devices at the ports, assigning resources to the ports".

In other words, claim 13 recites that criteria are set for transactions at the port with respect to the number of devices being serviced by the port, and resources are assigned to the ports based on the number of devices being serviced thereby.

Westerinen fails to teach or suggest such claim elements. Westerinen describes a system for auto-configuring a personal computer. See Abstract and Col. 2, lines 41-43. The system of Westerinen first determines what resources are available at the PC, such as interrupt request lines, direct memory access (DMA) channels, memory or input/output address space, etc. See Col. 1, lines 25-26. The system then determines what devices need access to those resources. Exemplary devices include modems, scanners, external storage, sound/video cards, network adapters, printers, SCSI ports, keyboard, mouse, etc. See Col. 1, limes 20-22. Finally, the system uses a predefined set of Rules to assign devices to available resources so as to achieve an efficient allocation of resources. See Col. 2, lines 46-57.

The Office Action cites to Col. 2, limes 43-53 and Col. 7, lines 21-33 of Westerinen as teaching or suggesting Applicants' claimed "setting criteria for transactions at the port with respect to the number of devices". Applicants respectfully disagree.

The cited excerpt from Col. 2 of Westerinen states as follows:

> A goal is to correctly configure a computer platform supporting ISA, EISA and/or PCI add-in adapters, and tune the resource assignments for performance. Knowledge on how to configure a system is gathered from sources such as experts who perform computer configurations and providers of devices and operating systems that are used in computers. From this information, Rules are generated which define the configuration process. The Rules based configuration program is generated by functionally grouping the rules into executable tasks such as "configuration" or "analysis" rules.

As shown, this excerpt describes how the rules, which Westerinen uses to configure and tune the PC's ISA, EISA and PCI adapters, are defined. Specifically, the excerpt discloses that the rules are derived from historical configurations created by experts. The excerpt provides no teaching or suggestion for the setting of criteria for transactions at the ports with respect to the numbers of devices being serviced by the ports, as recited by claim 13. For example, there is no mention in this excerpt of "transactions" at ports, nor the setting of criteria for these transactions with respect to the numbers of devices being serviced by the port.

The cited excerpt from Col. 7 of Westerinen states as follows:

> For each of the IRQ, DMA, and I/O port resources a list is preferably created and ordered in the following manner. A list with two groups, EISA/ISA Devices and PCI Devices is created, with the EISA/ISA group first in the list. The selection to assign PCI devices last stems from the fact that EISA and ISA adapters typically have fewer resource options than PCI Devices. The EISA/ISA Devices are ordered from the least number of options to the one with the most options. This ordering occurs only for the IRQ and DMA Resources. The PCI Devices are ordered according with the first unique instance of a PCI device grouped in the beginning of the PCI List (unique device determined using PCI Vendor ID and PCI Device ID).

This excerpt also fails to teach or suggest the claimed feature. Here, Westerinen describes how the subject computer's EISA/ISA and PCI devices are assigned to its

available IRQ, DMA and I/O port resources. In particular, for the IRQ, DMA and I/O port resources, lists are made of the EISA/ISA and PCI devices that seek access to those resources. The lists are then ordered depending on how many options the various devices can support. As was the case before, this excerpt similarly fails to teach or suggest the setting of criteria for transactions at the ports with respect to the numbers of devices being serviced by the ports, as recited by claim 13. Furthermore, the excerpt makes no mention of the setting any criteria at the ports for transactions at those ports, nor does it mention the setting of criteria based on the number of devices being serviced by the ports.

The Office Action next cites to Col. 8, lines 25-37 of Westerinen as teaching or suggesting Applicants' claimed, "with respect to the number of devices at the ports, assigning resources to the ports". Applicants again respectfully disagree.

This excerpt of Westerinen states as follows:

> With respect to Rules 5 and 6 regarding sharing resource values, during the assignment process, all potential setting values must be checked to verify that another device has not been allocated the same value. If the value is already being used and the resource can not be shared between the devices, then the current device is prohibited from using that value. The implementation of this checking mechanism is straight forward for those resources that use a single integer value, such as IRQ's and DMA's. If there is another device using a setting value and both resources cannot share values, then a conflict exists. Checking conflicts for resources that use a range of values (memory and I/O port) requires a determination as to whether there is overlap between the ranges.

This excerpt teaches that, during the configuration process, care must be taken not to assign the same setting value for a given resource to more than one device. The excerpt fails to teach or suggest Applicants' claimed limitation. Indeed, there is no mention at all for somehow assigning resources with respect to the number of devices at the ports.

Wilson, on which the Office Action also relies in combination with Westerinen, also fails to teach or suggest the limitations of claim 13. Wilson is directed to a system for reducing the number of interrupts generated when a plurality of devices all contend for access to a bus in order to transfer data to a processor. The Office Action, at p. 3, cites to Col. 10, lines 29-30 and 34-42 of Wilson as teaching Applicants' "determining the number of devices being serviced via the ports". Applicants respectfully disagree.

These excerpts from Wilson state as follows:

> FIG. 6 shows a flowchart of an exemplary method for generating reduced number of interrupts in accordance with another embodiment of the present invention. In this method, one or more I/O commands are received for transferring data between a host computer and one or more I/O devices in operation 602. Then, at each arbitration phase after a command completion, the host adapter monitors SCSI bus contention, in operation 604, to determine the number of devices arbitrating for the bus to re-select the host adapter. Next in operation 606, it is determined whether more than one device are contending for the bus. If so, data is transferred to the host adapter by a selected device having the highest priority (i.e., highest SCSI ID number) in operation 608.

Rather than determining the number of devices being serviced by some number of ports, this excerpt from Wilson teaches that its host adapter (316) determines how many SCSI devices (306) are trying to access the computer's primary bus (314) in order to issue an interrupt. See Fig. 3. There is no mention in this excerpt of somehow determining the number of devices being serviced by some number of ports. Indeed, if none of Wilson's SCSI devices (306) are arbitrating for the bus (314), then the determination that Wilson makes is zero, even though there are a number of SCSI devices (306) connected to the SCSI adapter (316). See Fig. 3.

Independent claim 15, in relevant part, recites as follows:

> "A system for programmably allocating resources to accommodate I/O transactions at I/O ports of a multiprocessor computer system, the system comprising:"
>
> "means for determining the number of devices being serviced via a port",
>
> "means for setting criteria for transactions **at the port** with respect to the number of devices", and
>
> "means, responsive to the criteria, for assigning resources to the ports".

As with the rejection of claim 13, the Office Action, at p. 4, similarly cites Col. 2, lines 43-53 and Col. 7, lines 21-33 of Westerinen as teaching or suggesting Applicants' claimed "means for setting criteria for transactions at the port with respect to the number of devices". The Office Action similarly cites Col. 8, lines 25-37 of Westerinen as teaching or suggesting Applicants' claimed "means, responsive to the criteria, for assigning devices to the ports".

As set forth above, the cited excerpts of Westerinen fail to teach or suggest these limitations. Instead, the cited references teach (1) how the rules, which Westerinen uses to configure and tune the PC's ISA, EISA and PCI adapters, are defined, (2) how the subject computer's EISA/ISA and PCI devices are assigned to its available IRQ, DMA and I/O port resources, and (3) that, during the configuration process, care must be taken not to assign the same setting value for a given resource to more than one device.

The Office Action, at p. 5, also relies on the Col. 10, lines 29-30 and 34-42 of Wilson, as teaching Applicants' claimed "means for determining the number of devices being serviced via a port". However, as set forth above, this excerpt from Wilson teaches

that its host adapter determines how many SCSI devices are trying to access the computer's primary bus in order to issue an interrupt.

Independent claim 21 in relevant part recites:

> "A method for programmably allocating resources for processing Input/Output (I/O) transactions at a plurality of I/O ports of an I/O bridge, the method comprising:"
>
> "identifying the number of I/O devices being serviced by at least one I/O port"
>
> "setting criteria for the transactions at the at least one I/O port with respect to the number of I/O devices being serviced by the port", and
>
> "assigning the resources to the at least one I/O port in response to the criteria".

Just as with the rejection of claims 13 and 15, the Office Action, at p. 6, once again cites Col. 2, lines 43-53 and Col. 7, lines 21-33 of Westerinen as teaching or suggesting Applicants' claimed "setting criteria for transactions at the port with respect to the number of devices", and to Col. 8, lines 25-37 of Westerinen as teaching or suggesting Applicants' claimed "assigning the resources to the at least one I/O port in response to the criteria".

For the reasons set forth above, the art of record fails to teach or suggest all of the limitations of claims 13, 15 and 21. Specifically, the art of record fails to teach or suggest determining the number of devices being serviced by the ports, setting criteria for transactions at the port with respect to the number of devices and, with respect to the numbers of devices at the ports, assigning resources to the ports. Because the art of record fails to teach or suggest all of the claim limitations, the rejections of claims 13, 15 and 21 should be reversed.

For the foregoing reasons, Applicants request that the rejection of independent claims 13, 15 and 21 be reversed.

Claim 32

Independent claim 32, in its entirety, recites as follows:

> "An Input/Output (I/O) bridge for use in a computer system having a plurality of processors, the I/O bridge comprising:"
>
> "a plurality of I/O ports, each I/O port configured to communicate with at least one I/O device that generates or receives transactions;"
>
> "resources for use in servicing the transactions of the I/O devices; and"
>
> "programmable logic configured and arranged to assign the resources **among the I/O ports** in response to the number of I/O devices with which the I/O ports are communicating."

As shown, claim 32, among other things, recites programmable logic disposed at an I/O bridge that is configured "to assign the resources **among I/O ports**" on the basis of the number of I/O devices with which the I/O ports are communicating.

The Office Action, at p. 11, cites to Col. 1, line 66 to Col. 2, line 12 of Westerinen as teaching Applicants' claimed "programmable logic" limitation. Applicants respectfully disagree.

The cited excerpt of Westerinen, in its entirety, states as follows:

> One embodiment of the present invention includes a self-configuring computer apparatus that comprises: processing logic that performs control and data processing functions; a plurality of resources that have resource settings therefore that are coupled to said processing logic and including at least interrupts and memory; a plurality of devices which require access to one or more of said plurality of resources; and **configuration logic that assigns**, substantially in parallel, **two or more of said plurality of devices to said resources**. Another embodiment of the present invention includes similar features and configuration logic that decides the order in which two or more of said plurality of devices are assigned to one of said

> plurality of resources before making an assignment of those two of more devices to that resource.

As shown, Westerinen discloses configuration logic that assigns devices directly to resources. This is not, however, what claim 32 recites. Claim 32 recites programmable logic configured to assign resources **among I/O ports** to which the devices, in turn, may be coupled.

In contrast, what Westerinen teaches is the assignment **of devices** directly to the resources, so that each device is assigned to a particular resource. There is no teaching or suggestion by Westerinen to assign resources **among I/O ports**. Instead, Westerinen operates solely at the device level, and provides no teaching or suggestion for assigning resources among I/O ports.

Because Westerinen fails to teach or suggest logic assigning resources among I/O ports, the obviousness rejection of claim 32 should be reversed.

Dependent claims 14, 16 and 31

In the Office Action, at pp. 5 and 10, claims 14, 16 and 31 were rejected as obvious based on Wallach. Specifically, the Office Action cites to Col. 10, lines 57-61 of Wallach. Applicants respectfully disagree that Wallach teaches or suggests the invention set forth in these dependent claims.

Claim 14, which depends from claim 13, recites as follows:

> "wherein assigning resources to the ports comprises at least one of assigning control registers to the ports, assigning direct memory access engines to the ports, assigning cache memory to the ports and assigning priorities among the transactions at the ports."

The excerpt from Wallach on which the final Office Action relies as purportedly teaching or suggesting this limitation reads as follows:

> "Once an adapter 310 is added to the computer system, system resources must be allocated for the adapter 310. The configuration manager 500 then configures the newly added adapter 310 (state 604) by writing information to the adapter's configuration space registers."

As shown, this excerpt discloses that, when a new adapter is added to Wallach's computer system, a configuration manager configures the new adapter for operation by writing certain information to the adapter's registers. Applicants respectfully submit that this excerpt provides no teaching or suggestion for the assignment of registers (as a resource) to the I/O ports of a multiprocessor computer system. Indeed, there is no mention by Wallach of somehow assigning registers or any other resource to the ports. Instead, the excerpt simply provides that information must be written to the new adapter's registers in order to configure it. Because Wallach fails to teach or suggest the "assigning of control registers **to the ports**", the rejection of claim 14 should be reversed.

Claim 16 depends from claim 15 and is similar to claim 14. Accordingly, for the reasons set forth above with regard to claim 14, among others, the rejection of claim 16 should also be reversed.

Claim 31, which depends from claim 21, recites as follows:

> "the I/O bridge comprises at least one control register, the at least one control register having a plurality of fields, and at least one field of the control register being associated with a corresponding resource, and"
>
> "the method further comprising **writing to a selected field** of the at least one control register **so as to modify the assignment of resources**."

As shown above, the Wallach excerpt, on which the Office Action similarly relies for rejecting claim 31, fails to teach or suggest writing to a selected field of a control register of an I/O bridge in order to modify the assignment of resources to an I/O port. Instead, at most, Wallach discloses that its adapter has registers which can be written to so as to configure the adapter. Again, there is no teaching or suggestion in this excerpt for providing a field of a register which can be written to in order to modify the assignment of resources to an I/O port. Because the cited reference fails to teach or suggest this limitation of claim 31, the rejection should be reversed.

Dependent claims 17 and 19

In the Office Action, claims 17 and 19 were rejected as obvious based on Westerinen. Applicants respectfully submit that Westerinen fails to teach or suggest the invention of claims 17 and 19.

Claim 17, in relevant part, recites:

> "determining the number and types **of transactions anticipated at the ports**, wherein the assignment of resources is further with respect to the numbers and types of transactions at the ports".

Westerinen at Col. 5, lines 38-46, on which the Office Action relies, states as follows:

> "The Configuration Ruleset is used to make resource assignments for a selected resource type (IRQ, DMA, etc.). The Agendas for the Configuration Ruleset are run in parallel using Agenda priority (discussed below) to determine the Rule's execution flow. For example, all active Rules in the "FindConflicts" Agenda are fired before any active Rules in the "Assign" Agenda. This provides a "parallel" implementation. The Configuration Ruleset preferably includes the Agendas of (in order of priority):"

In other words, what Westerinen is describing here is that, to generate the desired configuration, the rules' execution flow is determined using agendas that organize the

rules by types, such "FindConflicts" rules, "Assign" rules, etc. Applicants submit that this excerpt of Westerinen fails to teach or suggest determining the number and types of transactions anticipated at the ports wherein the assignment of resources to ports is based on the numbers and types of transactions anticipated at the ports. There is no mention of ports, or of somehow anticipating the number and types of transactions anticipated at the ports.

Claim 19, which depends from claim 15, is similar to claim 17, and was also rejected based on this same excerpt of Westerinen.

As set forth above, Westerinen fails to teach or suggest determining the number and types of transactions anticipated at the ports, or the assignment of resources with respect to the numbers and types of transactions at the ports, the rejections of claims 17 and 19 should be reversed.

CONCLUSION

Applicants respectfully submit that the claims are allowable over the art of record.

Accordingly, Applicants request that the rejection of all claims be reversed.

Respectfully submitted,

Michael R. Reinemann
Reg. No. 38,280
Tel. 617-951-3060

Please direct all correspondence to:

IP Administration Legal Department,
M/S 35
Hewlett-Packard Co.
P.O. Box 272400
Fort Collins, CO 80527-2400

CLAIMS APPENDIX
(Claims on Appeal in Appl. Ser. No. 09/944,776)

Claims 1-12 (Canceled)

13. (Previously presented) A method for programmably allocating resources to accommodate I/O transactions at I/O ports of a multiprocessor computer system comprising:

determining the number of devices being serviced via the ports,

identifying at least one assembly for hot swapping,

copying the contents of cache memories associated with the at least one identified assembly,

setting criteria for transactions at the port with respect to the number of devices, and

with respect to the numbers of devices at the ports, assigning resources to the ports.

14. (Previously presented) The method as defined in claim 13 wherein assigning resources to the ports comprises at least one of assigning control registers to the ports, assigning direct memory access engines to the ports, assigning cache memory to the ports and assigning priorities among the transactions at the ports.

15. (Previously presented) A system for programmably allocating resources to accommodate I/O transactions at I/O ports of a multiprocessor computer system, the system comprising:

means for determining the number of devices being serviced via a port,

at least one assembly identified for hot swapping,

means for copying the contents of cache memories associated with the at least one identified assembly,

means for setting criteria for transactions at the port with respect to the number of devices, and

means, responsive to the criteria, for assigning resources to the ports.

16. (Previously presented) The system as defined in claim 15 wherein the resources assigned to the ports comprises at least one of

direct memory access (DMA) engines,

cache memory, and

means for assigning priorities among the transactions at the ports.

17 (Previously presented) The method as defined in claim 13 further comprising determining the number and types of transactions anticipated at the ports, wherein the assignment of resources is further with respect to the numbers and types of transactions at the ports.

18. (Previously presented) The method as defined in claim 13 wherein the at least one identified assembly has a memory system, and the method further comprises copying the states and status of the memory systems associated with at least one identified assembly.

19. (Previously presented) The system as defined in claim 15 further comprising means for determining the number and types of transactions anticipated at the ports, wherein the criteria further accounts for the anticipated number and types of transactions.

20. (Previously presented) The system as defined in claim 15 wherein the at least one identified assembly has a memory system, and the system further comprises means for copying the states and status of the memory systems associated with the at least one identified assembly.

21. (Previously presented) A method for programmably allocating resources for processing Input/Output (I/O) transactions at a plurality of I/O ports of an I/O bridge, the method comprising:

identifying the number of I/O devices being serviced by at least one I/O port;

setting criteria for the transactions at the at least one I/O port with respect to the number of I/O devices being serviced by the port; and

assigning the resources to the at least one I/O port in response to the criteria.

22. (Previously presented) The method of claim 21 wherein the assigning comprises assigning a plurality of direct memory access (DMA) engines for use in processing I/O transactions.

23. (Previously presented) The method of claim 22 wherein assigning comprises apportioning a selected number of DMA engines to process a given transaction at a particular I/O port.

24. (Previously presented) The method of claim 22 wherein assigning comprises apportioning at least one DMA engine to process at least one transaction at a port.

25. (Previously presented) The method of claim 22 wherein assigning comprises apportioning one DMA engine to process a given transaction at a port identified as servicing multiple I/O devices.

26. (Previously presented) The method of claim 21 wherein assigning comprises assigning at least one miss address file (MAF) value for processing I/O transactions.

27. (Previously presented) The method of claim 21 wherein assigning comprises assigning a plurality of miss address file (MAF) values for processing I/O transactions.

28. (Previously presented) The method of claim 27 further comprising reducing the assigned number of MAF values.

29. (Previously presented) The method of claim 21 wherein

the I/O bridge is configured to utilize a plurality of virtual channels to communicate with at least one processors of a multiprocessor computer system, and

the resources include flow control credits associated with each of the plurality of virtual channels.

30. (Previously presented) The method of claim 29 wherein assigning comprises setting the number of flow control credits associated with each virtual channel.

31. (Previously presented) The method of claim 21 wherein

the I/O bridge comprises at least one control register, the at least one control register having a plurality of fields, and at least one field of the control register being associated with a corresponding resource, and

the method further comprises writing to a selected field of the at least one control register so as to modify the assignment of resources.

32. (Previously presented) An Input/Output (I/O) bridge for use in a computer system having a plurality of processors, the I/O bridge comprising:

a plurality of I/O ports, each I/O port configured to communicate with at least one I/O device that generates or receives transactions;

resources for use in servicing the transactions of the I/O devices; and

programmable logic configured and arranged to assign the resources among the I/O ports in response to the number of I/O devices with which the I/O ports are communicating.

33. (Previously presented) The I/O bridge of claim 32 wherein

the resources comprise at least one direct memory access (DMA) engine configured to process the transactions, and

the programmable logic apportions the at least one of DMA engine to process at least one transaction at a given I/O port in response to the number of I/O devices coupled to the given I/O port.

34. (Previously presented) The I/O bridge of claim 32 wherein

the resources include a plurality of miss address file (MAF) values for use in requesting information from the computer system, and

the programmable logic sets the number of available MAF values.

35. (Previously presented) The I/O bridge of claim 32 wherein

the I/O bridge communicates with the computer system through a plurality of virtual channels,

the resources include a plurality of flow control credits associated with the virtual channels, and

the programmable logic assigns a number of flow control credits to each virtual channel.

36. (Previously presented) the I/O bridge of claim 35 wherein the virtual channels comprise a Request channel, a Read I/O channel, and a Write I/O channel.

37. (Currently amended) An Input/Output (I/O) ~~The I/O~~ bridge ~~of claim 33 further~~ for use in a computer system having a plurality of processors, the I/O bridge comprising:

a plurality of I/O ports, each I/O port configured to communicate with at least one I/O device that generates or receives transactions;

resources for use in servicing the transactions of the I/O devices; and

programmable logic configured and arranged to assign the resources among the I/O ports in response to the number of I/O devices with which the I/O ports are communicating

at least one cache for storing information, wherein, to hot-swap an assembly of the computer system, the programmable logic is configured to

disable the at least one DMA engine, and

flush the information from the at least one cache, wherein

the resources comprise at least one direct memory access (DMA) engine configured to process the transactions, and

the programmable logic apportions the at least one of DMA engine to process at least one transaction at a given I/O port in response to the number of I/O devices coupled to the given I/O port.

38. (Previously presented) The I/O bridge of claim 37 wherein the at least one cache is one of a write cache, a read cache and a translation look-aside buffer (TLB).

39. (Previously presented) The I/O bridge of claim 37 wherein the assembly is a processor.

40. (Previously presented) The I/O bridge of claim 33 wherein

the programmable logic comprises at least one control register associated with each I/O port, and

the at least one control register has a first field for apportioning the at least one DMA engine.

41. (Previously presented) The I/O bridge of claim 32 wherein the programmable logic re-assigns resources among the I/O ports dynamically while the I/O bridge continues to operate.

EVIDENCE APPENDIX

None.

RELATED PROCEEDINGS APPENDIX

None.